

0280

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

#2

IN RE APPLICATION OF: KYUNG ET AL

SERIAL NO.: 09/760,645

FILED: January 17, 2001

FOR: APPARATUS AND METHOD FOR VERIFYING A
LOGIC FUNCTION OF A SEMICONDUCTOR
CHIP

GROUP ART UNIT: unknown

EXAMINER: unknown

ATTY. REFERENCE: EM/KYUNG/6426

THE COMMISSIONER FOR PATENTS
Washington, D.C. 20231

Sir:

The below identified communication(s) or document(s) is(are) submitted in the above application or proceeding:

- | | |
|--|--|
| <input type="checkbox"/> Declaration | <input type="checkbox"/> Issue Fee Transmittal |
| <input checked="" type="checkbox"/> Priority Document (Korean App.
No. 63713) | <input type="checkbox"/> Check in the Amount of \$ _____ |
| <input type="checkbox"/> Formal Drawings | <input type="checkbox"/> |
| <input type="checkbox"/> Small Entity Declaration(s) | <input type="checkbox"/> |

☒ Please debit or credit **Deposit Account Number 02-0200** for any deficiency or surplus in connection with this communication. A duplicate copy of this sheet is provided for use by the Deposit Account Branch.

☐

1

BACON & THOMAS, PLLC
625 SLATERS LANE - FOURTH FLOOR
ALEXANDRIA, VIRGINIA 22314
(703) 683-0500

DATE: February 2, 2001

Respectfully submitted,



Eugene Mar
Attorney for Applicant
Registration Number: 25,893



대한민국 특허청
KOREAN INDUSTRIAL
PROPERTY OFFICE

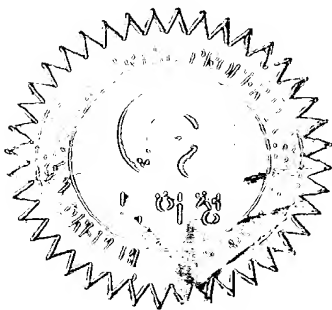
별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto
is a true copy from the records of the Korean Industrial
Property Office.

출원번호 : 특허출원 2000년 제 63713 호
Application Number

출원년월일 : 2000년 10월 28일
Date of Application

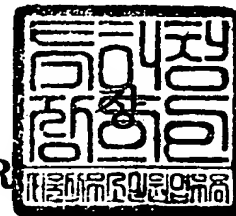
출원인 : (주)다이나렛시스템
Applicant(s)



2001 01 08
년 월 일

특 허 청

COMMISSIONER



【서류명】	출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	4626
【제출일자】	2000. 10. 28
【국제특허분류】	G06F
【발명의 명칭】	반도체 칩의 논리 기능 검증용 에뮬레이터 장치 및 방법
【발명의 영문명칭】	APPARATUS FOR EMULATING A LOGIC FUNCTION OF A SEMICONDUCTOR CHIP AND METHOD THEREOF
【출원인】	
【명칭】	(주)다이나믹시스템
【출원인코드】	1-2000-032888-0
【대리인】	
【성명】	주성민
【대리인코드】	9-1998-000517-7
【대리인】	
【성명】	장수길
【대리인코드】	9-1998-000482-8
【발명자】	
【성명의 국문표기】	경종민
【성명의 영문표기】	KYUNG, Chong-Min
【주민등록번호】	530621-1000823
【우편번호】	305-333
【주소】	대전광역시 유성구 어은동 99 한빛아파트 135-901
【국적】	KR
【발명자】	
【성명의 국문표기】	기안도
【성명의 영문표기】	KI, Ando
【주민등록번호】	641222-1119822
【우편번호】	305-333
【주소】	대전광역시 유성구 어은동 99 한빛아파트 121-501
【국적】	KR

【발명자】

【성명의 국문표기】 이승종
【성명의 영문표기】 LEE, Seung-Jong
【주민등록번호】 720117-1052828
【우편번호】 411-372
【주소】 경기도 고양시 일산구 주엽2동 문촌마을 1807동 1001호
【국적】 KR

【발명자】

【성명의 국문표기】 전영욱
【성명의 영문표기】 JEON, Young-Wook
【주민등록번호】 720125-1058327
【우편번호】 302-120
【주소】 대전광역시 서구 둔산동 909 수정타운 16/806
【국적】 KR

【심사청구】

청구

【취지】

특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사를 청구합니다. 대리인
 주성민 (인) 대리인
 장수길 (인)

【수수료】

【기본출원료】	20 면	29,000 원
【가산출원료】	27 면	27,000 원
【우선권주장료】	0 건	0 원
【심사청구료】	40 항	1,389,000 원
【합계】		1,445,000 원
【감면사유】	소기업 (70%감면)	
【감면후 수수료】		433,500 원

【첨부서류】

1. 요약서·명세서(도면)_1통 2. 소기업임을 증명하는 서류_1통[사업자등록증 사본] 3. 소기업임을 증명하는 서류_1통[원천징수이행상황신고서 원본] 4. 소기업임을 증명하는 서류_1통[집합건축물대장 원본] 5. 위임장_1통[원본]

【요약서】

【요약】

본 출원에는 프로세싱 엔진과 타겟 인터페이스 엔진이 상호 작용하는 에뮬레이션 환경에서 프로세싱 엔진이 수행하는 소프트웨어 모델의 변수 값과 타겟 인터페이스 엔진에 의해 생성되는 핀 신호를 동기시켜서 관찰할 수 있도록 해 주는 개선된 에뮬레이터 장치 및 방법이 개시되어 있다. 타겟 시스템에 장착될 타겟 칩의 논리 설계를 검증하기 위한 본 발명에 따른 에뮬레이터 장치는 상기 타겟 칩의 논리 설계에 대응하는 적어도 하나의 소프트웨어 변수를 포함하는 소프트웨어 알고리즘을 실행하기 위한 프로세싱 엔진, 및 상기 타겟 시스템과 인터페이스하고 있으며, 상기 프로세싱 엔진의 알고리즘 출력에 응답하여 상기 타겟 시스템으로 또는 상기 타겟 시스템으로부터 핀 신호를 송수신하기 위한 타겟 인터페이스 엔진을 포함한다. 상기 적어도 하나의 소프트웨어 변수와 상기 핀 신호는 상기 알고리즘의 실행에 따라 시간적으로 변화하며, 상기 프로세싱 엔진은 선정된 시간에서의 상기 적어도 하나의 소프트웨어 변수 값과 상기 핀 신호 값을 대응시키기 위한 수단을 포함하고 있다. 또한, 타겟 시스템에 장착될 타겟 칩의 논리 설계를 검증하기 위한 본 발명에 따른 에뮬레이션 방법은 상기 타겟 칩의 논리 설계에 대응하는 적어도 하나의 소프트웨어 변수를 포함하는 소프트웨어 알고리즘을 실행하는 단계, 상기 소프트웨어 알고리즘의 실행에 따라 상기 타겟 시스템으로 전송될 핀 신호를 생성하는 단계, 및 상기 적어도 하나의 소프트웨어 변수와 상기 핀 신호의 변화 값에 대한 시간적 이력을 제공하는 단계를 포함한다.

【대표도】

도 4

【색인어】

에뮬레이터, 핀 신호 관찰기, 소프트웨어 변수 관찰기

【명세서】**【발명의 명칭】**

반도체 칩의 논리 기능 검증용 에뮬레이터 장치 및 방법{APPARATUS FOR EMULATING A LOGIC FUNCTION OF A SEMICONDUCTOR CHIP AND METHOD THEREOF}

【도면의 간단한 설명】

도 1은 본 발명에 따른 에뮬레이터 장치와 타겟 시스템이 상호 연결된 전체 시스템 구성을 도시한 블록도.

도 2는 본 발명의 바람직한 실시예에 따른 프로세싱 엔진의 구성을 도시한 블록도.

도 3은 본 발명의 바람직한 실시예에 따른 타겟 인터페이스 엔진의 구성을 도시한 블록도.

도 4는 본 발명의 바람직한 실시예에 따른 프로세싱 엔진 내부의 메모리의 소프트웨어 구성을 도시한 블록도.

도 5는 본 발명의 바람직한 실시예에 따른 핀 신호 관찰기의 세부 구성을 도시한 블록도.

도 6은 본 발명의 바람직한 실시예에 따른 신호 선택기의 세부 구성을 도시한 블록도.

도 7은 본 발명의 바람직한 실시예에 따른 신호 샘플러의 구성을 도시한 블록도.

도 8은 본 발명의 바람직한 실시예에 따른 트리거 사건 감지기의 구성을 도시한 블록도.

도 9는 본 발명에 따라 핀 신호 관찰기측에서 감지된 하드웨어 트리거 사건 발생에 응답하여 핀 신호 관찰기측에서 일어나는 처리 과정을 도시한 플로우차트.

도 10은 본 발명에 따라 핀 신호 관찰기측에서 감지된 하드웨어 트리거 사건 발생에 응답하여 프로세싱 엔진측에서 일어나는 처리 과정을 도시한 플로우차트.

도 11은 본 발명에 따라 타겟 칩 알고리즘내에서의 소프트웨어 트리거 사건 발생에

응답하여 프로세싱 엔진측에서 일어나는 처리 과정을 도시한 플로우차트.

도 12는 종래 기술에 있어서 소프트웨어 변수 관찰 값과 하드웨어 핀 신호 관찰 값 사이의 측정 지연 시간을 도시한 타이밍도.

도 13은 본 발명에 따라 소프트웨어 변수 값의 저장 손실이 방지되는 원리를 도시한 도면.

【발명의 상세한 설명】

【발명의 목적】

【발명이 속하는 기술분야 및 그 분야의 종래기술】

<14> 본 발명은 일반적으로 반도체 칩 에뮬레이션 기법에 관한 것으로, 특히, 반도체 칩의 논리 설계에 대응하는, 고급 언어로 구현된 소프트웨어 알고리즘 또는 모델에 기초하여 반도체 칩의 논리 기능을 검증하기 위한 에뮬레이터 장치 및 그 방법에 관한 것이다.

<15> 집적 회로로 이루어진 반도체 칩의 제조에는 많은 시간과 비용이 소요될 뿐 아니라, 최근에는 이들 반도체 칩 및 이를 탑재한 시스템의 복잡도가 점차 높아지

고 있다. 따라서, 이들 칩을 실제 제조하기 이전의 설계 과정에서부터 설계 오류를 제거하는 것이 바람직하고, 이에 따라 칩의 '논리 설계(logic design)'를 검증하고자 하는 많은 노력이 있어 왔다. 그러한 검증 방법 중의 하나로, 반도체 칩 (이하, '타겟 칩'이라 함)의 타겟 논리 (타겟 칩으로 설계 및 구현되어야 할 논리 기능)의 실제 구현에 앞서 타겟 논리의 기능적 오류를 타겟 시스템 (타겟 칩이 장착될 응용 시스템)과 연관 지어 검증하는 에뮬레이션 방법이 알려져 있다.

<16> 통상적으로, 논리 기능의 검증은 소프트웨어 모델링, 하드웨어 모델링 또는 이들의 조합에 의해 수행되어져 왔다. 소프트웨어 모델링이란 타겟 칩에 의해 수행될 논리 기능을 소프트웨어를 통해 기술하는 것을 말하는 것으로, 이에선 주로 C, C++ 등과 같은 고급 프로그래밍 언어나 VHDL, Verilog 등과 같은 하드웨어 기술 언어(HDL)가 사용되어 왔다. 그러나, 하드웨어 기술 언어는 C, C++ 등의 소프트웨어 프로그래밍 언어에 비해 타겟 칩의 논리 기능을 기술하거나 디버그하기가 용이하지 않으며 그 실행 속도 또한 느리기 때문에, 논리 기능의 설계 및 검증에 많은 시간이 소요되는 문제점이 있다. 이에, 최근에는 타겟 칩의 논리 기능을 C, C++ 등의 소프트웨어 프로그래밍 언어를 사용하여 모델링한 뒤, 이를 타겟 시스템에 접목해 그 기능적 오류를 검증하고, 이어서 이 논리 기능을 하드웨어와 소프트웨어로 각각 분할하여 구현하는 형태의 논리 기능 설계 및 검증 방식이 주목 받고 있다. 이러한 접근 방식은 전술한 하드웨어 기술 언어(HDL) 기반의 에뮬레이션 방식에 비해 설계 및 검증에 소요되는 시간을 현저히 줄일 수 있으므로, 최근 등장한 단일 칩 시스템(System-On-A-Chip) 등에서의 타겟 칩의 논리 기능의 복잡도가 증가하는 경우에 보다 효율적인 에뮬레이션을 제공할 수 있다.

<17> 한편, 미국특허 제4,901,259호는 소프트웨어 프로그래밍 언어 또는 HDL로 기술된

소프트웨어 모델을 인터페이스를 통해 타겟 시스템에 연결된 범용 컴퓨터를 사용하여 실행하는 'ASIC 에뮬레이터'를 개시하고 있다. 그러나, 이 ASIC 에뮬레이터는 단순히 범용의 호스트 컴퓨터에서 실행되고 있는 ASIC 소프트웨어 모델에 의해 생성된 핀 I/O와 타겟 시스템상의 하드웨어 장치에 의해 생성된 핀 I/O를 서로 동기시키는 일반적인 핀 I/O 동기화기에 불과하다. 즉, 타겟 시스템과의 외부 인터페이스용으로 사용되는 모든 핀 신호 값들이 호스트 컴퓨터인 범용 컴퓨터에서 실행되고 있는 ASIC 소프트웨어 모델에 의해 생성되고, ASIC 에뮬레이터는 단지 이 핀 신호 값들을 전기적 신호로 변환하여 타겟 시스템내의 ASIC bare 소켓으로 전달하는 역할만을 수행한다. 그러나, 호스트 컴퓨터내의 소프트웨어 모델은 이러한 외부 인터페이스용 핀 신호들의 생성과 관련된 전 과정을 실행하기에는 그 실행 속도가 너무 느리고, 또한 호스트 컴퓨터 역시 범용 오퍼레이팅 시스템에 그 기반을 두고 있기 때문에 논리 기능의 알고리즘의 실행에만 컴퓨팅 자원을 할당할 수 없어 결과적으로 ASIC 소프트웨어 모델의 알고리즘의 실행 속도가 늦어진다는 문제점이 있다. 따라서, 이러한 에뮬레이팅 시스템을 통해서만 고속 에뮬레이션이 불가능하고, 다양한 VLSI 설계에 어려움이 따른다

<18> 본 발명의 발명자는 이상의 종래 에뮬레이팅 시스템의 문제점에 대처하기 위하여, 미국특허출원 제09/680,467호에서, 타겟 시스템과 연동시킨 상태의 시스템 레벨에서 C, C++ 등의 고급 프로그래밍 언어로 모델링 된 타겟 칩의 논리 기능을 설계의 초기 단계에서 고속 검증할 수 있도록 하는 에뮬레이터 장치 및 방법을 제안한 바 있다. 이러한 에뮬레이터 장치는, 고급 프로그래밍 언어로 작성된 알고리즘을 컴파일하여 얻은 코드를 실행하는 적어도 하나의 프로세서를 포함하는 프로세싱 엔진과, 타겟 시스템과의 외부 인터페이스가 가능한 핀 신호를 생성하는 Field-Programmable Gate Arrays(FPGAs) 또는

Programmable Logic Devices(PLDs)와 같은 적어도 하나의 재설정가능 논리 장치를 포함하는 타겟 인터페이스 엔진의 두 종속 모듈로 이루어진다.

<19> 그러나, 전술한 미국특허출원 제09/680,467호에 개시된 에뮬레이터 장치에서는, 관측 대상인 하드웨어 핀 신호 또는 소프트웨어 변수값을 논리 분석기 또는 소프트웨어 디버거를 사용하여 디버깅을 수행하는 종래의 에뮬레이션 환경에서의 디버깅 방법과는 달리, 두 종속 모듈이 각기 다른 속도로 동작할 뿐만 아니라 두 모듈간의 상호 작용을 위한 인터페이스로 인해 두 모듈간의 정보 교환에 지연이 필연적으로 발생하기 때문에 이들 각 모듈에서 관찰되는 하드웨어 핀 신호의 파형과 소프트웨어 변수 값을 동기시켜 관찰해야만 한다는 문제점이 있다.

<20> 따라서, 이상의 프로세싱 엔진과 타겟 인터페이스 엔진이 상호 작용하는 에뮬레이션 환경에서 타겟 칩의 논리 기능을 설계하고 검증하는데 있어서는, 프로세싱 엔진이 수행하는 소프트웨어 모델내의 변수 값과 타겟 시스템과 인터페이스하고 있는 타겟 인터페이스 엔진으로부터의 하드웨어 핀 신호의 파형을 동기시켜 관찰할 수 있도록 해주는 메커니즘이 절실히 요구되고 있다.

【발명이 이루고자 하는 기술적 과제】

<21> 따라서, 본 발명의 목적은 목표 칩의 논리 설계를 용이하게 디버깅할 수 있도록 하여 에뮬레이션 사용 환경을 개선한 개선된 에뮬레이터 장치 및 방법을 제공하는 것이다.

<22> 본 발명의 다른 목적은 프로세싱 엔진과 타겟 인터페이스 엔진이 상호 작용하는 에뮬레이션 사용 환경에서 프로세싱 엔진이 수행하는 소프트웨어 모델의 변수 값과 타겟 인터페이스 엔진에 의해 생성되는 핀 신호 값의 시간적 변화를 용이하게 추적할 수 있도록

록 해 주는 개선된 에뮬레이터 장치 및 방법을 제공하는 것이다.

<23> 본 발명의 또 다른 목적은 프로세싱 엔진과 타겟 인터페이스 엔진이 상호 작용하는 에뮬레이션 사용 환경에서 프로세싱 엔진이 수행하는 소프트웨어 모델의 변수 값과 타겟 인터페이스 엔진에 의해 생성되는 핀 신호를 동기시켜서 관찰할 수 있도록 해 주는 개선된 에뮬레이터 장치 및 방법을 제공하는 것이다.

<24> 본 발명의 한 특징에 따르면, 타겟 시스템에 장착될 타겟 칩의 논리 설계를 검증하기 위한 에뮬레이터 장치가 제공되는데, 이 장치는 상기 타겟 칩의 논리 설계에 대응하는 적어도 하나의 소프트웨어 변수를 포함하는 소프트웨어 알고리즘을 실행하기 위한 프로세싱 엔진, 및 상기 타겟 시스템과 인터페이스하고 있으며, 상기 프로세싱 엔진의 알고리즘 처리에 응답하여 상기 타겟 시스템으로 또는 상기 타겟 시스템으로부터 핀 신호를 송수신하기 위한 타겟 인터페이스 엔진을 포함한다. 또한, 상기 적어도 하나의 소프트웨어 변수와 상기 핀 신호는 상기 알고리즘의 실행에 따라 시간적으로 변화하며, 상기 프로세싱 엔진은 선정된 시간에서의 상기 적어도 하나의 소프트웨어 변수 값과 상기 핀 신호 값을 대응시키기 위한 수단을 포함하고 있다.

<25> 본 발명의 한 실시예에 따르면, 상기 타겟 인터페이스 엔진은 상기 타겟 시스템으로 송신될 핀 신호를 생성하며 상기 타겟 시스템으로부터 핀 신호를 수신하기 위한 핀 신호 생성기와, 상기 핀 신호 생성기와 상기 타겟 시스템 사이에서 송수신되는 상기 핀 신호를 입력받아 저장하기 위한 핀 신호 관찰기를 포함한다. 이 핀 신호 관찰기는 상기 핀 신호 값이 특정 조건을 만족시키는 경우를 정의하는 제1 트리거 사건의 발생을 감지하고, 상기 제1 트리거 사건의 발생에 응답하여 상기 프로세싱 엔진으로 제1 인터럽트 요청 신호를 전송한다. 이 제1 인터럽트 요청 신호를 수신한 상기 프로세싱 엔진은 상

기 핀 신호 관찰기에 저장된 핀 신호 값을 그 내부에 있는 핀 신호값 저장 버퍼에 저장한다.

<26> 본 발명의 다른 실시예에 따르면, 상기 프로세싱 엔진은 상기 시간에 따라 변화하는 적어도 하나의 소프트웨어 변수를 관찰하고, 관찰되는 소프트웨어 변수 값이 저장될 수 있도록 제어하는 소프트웨어 변수 관찰기를 포함한다. 이 소프트웨어 변수 관찰기는 상기 시간에 따라 변화하는 소프트웨어 변수의 값이 특정 조건을 만족시키는 경우를 정의하는 제2 트리거 사건의 발생을 감지하며, 상기 제2 트리거 사건의 발생에 응답하여 제2 인터럽트 요청 신호를 출력한다. 이 때, 프로세싱 엔진은 상기 제2 인터럽트 요청 신호에 응답하여 상기 핀 신호 관찰기에 저장된 핀 신호 값을 상기 핀 신호값 저장 버퍼에 저장한다.

<27> 본 발명의 다른 특징에 따르면, 타겟 시스템에 장착될 타겟 칩의 논리 설계를 검증하기 위한 에뮬레이션 방법이 제공되는데, 이 방법은 상기 타겟 칩의 논리 설계에 대응하는 적어도 하나의 소프트웨어 변수를 포함하는 소프트웨어 알고리즘을 실행하는 단계, 상기 소프트웨어 알고리즘의 실행에 따라 상기 타겟 시스템으로 전송될 핀 신호를 생성하는 단계, 및 상기 적어도 하나의 소프트웨어 변수와 상기 핀 신호의 변화 값에 대한 시간적 이력을 제공하는 단계를 포함한다.

<28> 본 발명의 한 실시예에 따르면, 상기 시간적 이력을 제공하는 단계는 상기 적어도 하나의 소프트웨어 변수의 시간에 따른 변화와 상기 핀 신호의 시간에 따른 변화를 기록하는 단계를 포함한다.

<29> 이러한 본 발명의 구성에 따르면, 에뮬레이션 장치의 두 종속 시스템인 프로세싱 엔진과 타겟 인터페이스 엔진 사이에서 발생하는 통신 시간 지연으로 인한 정보 손실을

보상하여, 소프트웨어 변수값과 이에 대응되는 하드웨어 핀 신호 값의 관찰 타이밍을 동기시킴으로써, 타겟 칩의 논리 설계 검증이 용이하게 된다.

【발명의 구성 및 작용】

<30> 도 1을 참조하면, 본 발명에 따른 에뮬레이터 장치와 타겟 시스템이 상호 연결된 전체 시스템 구성이 도시되어 있다. 도 1에 도시된 바와 같이, 본 발명에 따른 에뮬레이터 장치(100)는 프로세싱 엔진(101), 타겟 인터페이스 엔진(102), 그리고 이들을 상호 인터페이스하는, 예를 들어 33MHz 클럭으로 동작하는 PCI 버스 또는 Accelerated Graphics Port(AGP) 등과 같은 통신 채널(103)을 포함한다. 프로세싱 엔진(101)은 RS-232C 또는 USB 등과 같은 통신 포트(105)를 통해 호스트 컴퓨터(110)와 통신한다. 타겟 인터페이스 엔진(102)은 소켓 아답터(104)를 통해 에뮬레이션의 대상이 되는 타겟 칩이 장착 될 타겟 시스템(120)내의 소켓(121)에 결합된다. 소켓(121)은 타겟 시스템(120) 내에 있는 다른 로직(123) 또는 집적 회로(124)들과 연결되어, 타겟 시스템(120)의 일부 또는 전체 논리 기능을 수행하게 된다. 한편, 도 1에 도시한 바와 같이, 타겟 시스템(120) 내에는 에뮬레이션 대상이 되는 타겟 칩이 복수 개 존재할 수 있고, 이 경우 다른 타겟 칩은 타겟 시스템(120)상에 마련된 다른 소켓(122)을 통해 다른 에뮬레이터 장치(130)와 연결될 수 있다.

<31> 도 2는 도 1에 도시된 본 발명에 따른 프로세싱 엔진(101)의 일실시예를 도시한 도면이다. 도시된 바와 같이, 프로세싱 엔진(101)은 프로세서(201), 메모리(202), 호스트 브리지(203), 타이머(204), 및 기타 로직(205)으로 이루어진다. 프로세싱 엔진(101) 내의 메모리(202)에는 C/C++ 또는 이와 유사한 고급 언어로 작성된, 타겟 칩의 논리 기능에 대응하는 소프트웨어 알고리즘이 저장되는데, 실제로 이 알고리즘은 호스트

컴퓨터(110)에 의해 컴파일된 후, 실행 코드(executable code) 상태로 통신 포트(105)를 통해 프로세싱 엔진(101)의 메모리(202)로 다운로드되어 저장된다. 프로세싱 엔진(101) 내의 프로세서(201)는 이 다운로드된 실행 코드를 실행하여 타겟 칩의 특정 응용 분야에 대응하는 신호 처리, 변/복조, 암호화/복호화 등의 타겟 칩의 내부 논리 기능을 수행한다. 프로세서(201)와 메모리(202)는 호스트 브리지(203)를 통해 후술할 타겟 인터페이스 엔진(102)에 접속된다. 기타 로직(205)은 통신 포트(105)와 도시되지 않은 클럭 발생기, 전압 변환기 등 프로세싱 엔진(101)이 동작하는데 필요한 추가 회로들을 포함한다. 타이머(204)는 후술할 핀 신호와 소프트웨어 변수의 동기 관찰에 필요한 정확한 시간 정보를 제공하기 위한 것으로서, 그 일례로 카운터 등을 들 수 있다.

<32> 도 4는 도 2에 도시된 프로세싱 엔진(101) 내의 메모리(202)의 소프트웨어 구성을 도시한 도면이다. 도시된 바와 같이, 메모리(202)는 타겟 칩 알고리즘(400), 소프트웨어 변수 관찰기(401), 인터럽트 처리기(402), 타겟 인터페이스 엔진 제어기(403), 소프트웨어 변수값 저장 버퍼(404), 핀 신호값 저장 버퍼(405), 및 변수 및 신호 분석기(406)로 분할된다. 타겟 칩 알고리즘(400)은 전술한 바와 같이 호스트 컴퓨터(110)로부터 다운로드된 타겟 칩의 내부 논리 기능과 관련된 실행 코드가 저장되는 메모리 영역이다. 소프트웨어 변수 관찰기(401)는 관찰 대상이 되는 소프트웨어 변수 값의 변경이 발생하는 타겟 칩 알고리즘의 소스 코드 부분에 삽입되는 함수 루틴이 저장되는 메모리 영역으로서, 이 함수 루틴은 전술한 타겟 칩 알고리즘(400)과 마찬가지로 호스트 컴퓨터(110)에 의해 컴파일된 상태로 다운로드되어 저장된다.

<33> 이 함수 루틴은, 예를 들어 타겟 칩 알고리즘(400)에서 'A'라는 변수의 변화를 관찰하고자 하는 경우, 컴파일되기 이전의 소스 코드에서 변수 'A'와 관련되는 부분에 추가되

거나 변경된다. 즉, 소스 코드 중 변수 'A'에 값 'b'를 기록하라는 소스 코드 'A = b;'가 발견되면, 이를 'SW_MONITOR(&A, b);'라는 소프트웨어 루틴으로 변환한다. 본 발명의 바람직한 실시예에 따르면, 소프트웨어 루틴 'SW_MONITOR()'은 변수 'A'에 값 'b'를 기록하면서 동시에 메모리(202)에 별도로 마련된 소프트웨어 변수값 저장 버퍼(404)에 해당 변수의 변경 값을 저장하도록 하는 함수 루틴이 될 수 있다. 따라서, 프로세서(201)가 타겟 칩 알고리즘(400)의 수행 중 첨가된 이와 같은 함수 루틴(401)을 만나게 되면, 이 함수 루틴에 의해 관찰 대상이 되는 소프트웨어 변수 값이 소프트웨어 변수값 저장 버퍼(404)에 저장될 수 있게 된다. 이 경우, 후술할 타이머(204) (도 2 참조)에서 제공되는 시간-정보도 해당 변수 값과 함께 소프트웨어 변수값 저장 버퍼(404)에 저장될 수 있다. 또한, 소프트웨어 변수 관찰기(401)에는 타겟 칩 알고리즘(400) 내의 소프트웨어 변수들 중 임의의 선택된 변수들의 값의 소정 조합으로 정의되는, 사용자에게 의해 설정되는 소프트웨어 트리거 조건을 미리 지정해 둘 수 있는데, 이 소프트웨어 변수 관찰기(401)는 상기 트리거 조건이 만족되는 경우 소프트웨어적인 인터럽트를 발생하여 후술할 인터럽트 처리기(402)가 수행될 수 있도록 한다.

<34> 한편, 인터럽트 처리기(402)는 프로세싱 엔진(101)이 타겟 칩 알고리즘(400)을 수행하는 도중 소프트웨어 모듈에서 발생하는 인터럽트를 처리하는 루틴이 저장되는 메모리 영역이다. 인터럽트 처리기(402)는 프로세서(201)에 의한 알고리즘(400)의 수행 도중, 소프트웨어 변수 관찰기(401)에 의해 관찰되어 소프트웨어 변수값 저장 버퍼(404)에 저장되는 소프트웨어 변수 값이 전술한 소프트웨어 트리거 조건과 일치하게 될 경우, 이에 따른 인터럽트를 처리하게 된다. 또한, 인터럽트 처리기(402)는 이러한 소프트웨어적인 인터럽트의 처리뿐 아니라, 후술할 타겟 인터페이스 엔진(102)측에서 발생하는

핀 신호 값들의 조합으로 정의된 하드웨어적인 인터럽트에 대해서도 유사한 처리를 행한다. 인터럽트 처리기(402)에 의해 수행되는 상세 내용에 대해서는 이하에서 상술하기로 한다.

<35> 타겟 인터페이스 엔진 제어기(403)는 타겟 인터페이스 엔진(102)을 제어하는 루틴이 저장되는 메모리 영역으로, 전술한 인터럽트 처리기(402)의 인터럽트 처리에 응답하여, 타겟 인터페이스 엔진(102)내의 핀 신호 관찰기에 의해 관찰 및 저장된 핀 신호 값들을 불러와 이를 메모리(202)내의 핀 신호값 저장 버퍼(405)에 저장하는 역할을 한다.

<36> 변수 및 신호 분석기(406)는 소프트웨어 변수값 저장 버퍼(404) 및 핀 신호값 저장 버퍼(405)에 저장된 정보를 이용하여, 예를 들어 에뮬레이션의 진행 상태 및 오류를 분석하는 작업을 수행하도록 프로그램될 수 있다.

<37> 도 3은 본 발명에 따른 타겟 인터페이스 엔진(102)의 일실시예를 도시한 도면이다. 타겟 인터페이스 엔진(102)은 통신 채널 제어기(300), 핀 신호 생성기(301), 및 핀 신호 관찰기(302)로 구성된다. 타겟 인터페이스 엔진(102)의 핀 신호 생성기(301) 및 핀 신호 관찰기(302)는 통신 채널 제어기(300)의 제어에 따라 통신 채널(103)을 통해 프로세싱 엔진(101)과 통신하도록 되어 있다. 핀 신호 생성기(301)는 프로세싱 엔진(101)에서의 알고리즘 처리에 응답하여 타겟 시스템(120)으로 또는 타겟 시스템(120)으로부터 기록/판독/제어 신호 등의 외부 핀 신호를 송수신한다. 핀 신호 생성기(301)는 타겟 시스템(120)과 인터페이스하고 있으며, 타겟 시스템(120)의 응용에 맞추어 타겟 시스템(120)상의 논리 회로 및 IC 등과 상호 동작하도록 설정되는 논리 장치이다. 본 발명의 바람직한 실시예에 따르면, 핀 신호 생성기(301)는 FPGA, PLD 등과 같은 프로그램가능/재설정가능 칩에 의해 구현될 수 있다. 타겟 시스템(120)에 적합하게 구현된 핀 신호 생성

기(301)는 적절한 외부 핀 신호(104)들을 생성하여 이를 타겟 시스템(120)으로 제공함으로써, 결과적으로 타겟 시스템(120)의 응용이 실행되도록 한다.

<38> 핀 신호 관찰기(302)는 핀 신호 생성기(301)와 타겟 시스템(120) 간에 송수신되는 신호 및/또는 핀 신호 생성기(301) 내부의 신호를 입력으로 받아 저장하며, 전술한 핀 신호 생성기(301)와 마찬가지로 FPGA 또는 PLD등과 같은 재설정가능 또는 프로그램 가능한 논리 장치로 구현될 수 있다. 도 5를 참조하면, 본 발명의 바람직한 실시예에 따른 핀 신호 관찰기의 세부 구성이 예시되어 있다. 도시된 바와 같이, 핀 신호 관찰기(302)는 신호 선택기(501), 신호 샘플러(502), 트리거 사건 감지기(503), 버퍼(504), 클럭 발생 회로(506), 스탬퍼(507), 저장 메모리(505) 및 핀 신호 관찰기 제어기(500)로 구성된다. 신호 선택기(501)는 핀 신호 생성기(301)와 타겟 시스템(120) 간의 인터페이스 신호(104, 304)와 핀 신호 생성기(301) 내부 회로의 신호(303)를 입력받아, 이 중 사용자에게 의해 선택되는 관찰 대상의 신호만을 선택한다. 즉, 신호 선택기(501)는 핀 신호 관찰기(302)가 입력으로 받아들이는 다수의 핀 신호 중 관찰하고자 하는 일부 핀 신호만을 선택하는 역할을 하는 것으로, 이러한 신호 선택기(501)의 핀 신호 선택 기능은 사용자에게 의해 임의로 프로그램 또는 재설정될 수 있다.

<39> 도 6은 5개 입력(601)으로부터 3개의 출력(602) 신호를 선택하는 신호 선택기의 일 실시예를 도시하고 있다. 도시된 바와 같이, 사용자는 FPGA 또는 PLD등과 같은 재설정 가능칩의 프로그램 가능 영역(603)에 미리 정의되어 있는 연결을 개방(604) 또는 단락(605) 상태로 적절히 프로그램함으로써 다수의 입력 핀 신호

중 일부를 선택할 수 있다. 도6에서, 01은 I4, 02는 I2, 03는 I3가 연결되어 선택되는 예를 보이고 있다. 이상의 신호 선택기(501)를 이용함으로써, 사용자는 다수의 핀 신호들 중 관찰 대상의 핀 신호만을 부분적으로 선택할 수 있으며 아울러 이들 선택된 핀 신호들을 일정한 순서로 정렬할 수도 있게 된다.

<40> 다시 도 5를 참조하면, 신호 선택기(501)의 의해 선택된 핀 신호들은 계속하여 클럭 발생 회로(506)에서 생성된 클럭 신호에 기초하여 신호 샘플러(502)에 의해 샘플링된다. 도7은 본 발명에 따른 신호 샘플러(502)의 일실시예를 도시한 도면이다. 도시한 바와 같이 신호 샘플러(502)는 입력 신호, 즉 신호 선택기(501)에 의해 선택된 핀 신호(508)의 개수에 상응하는 복수 개의 플립플롭(701a-701n)으로 구성될 수 있다. 각 플립플롭(701a-701n)은 클럭 발생 회로(506)로부터 생성된 클럭 신호(510)에 동기되어 입력된 핀 신호(508)를 샘플링함으로써 출력 신호(509)를 생성한다. 이 때, 신호 샘플러(502)에 의해 검출된 샘플링 시점을 구별하기 위하여, 출력 신호(509)에는 스탬퍼(507)에 의해 생성된 시간 정보(511)가 함께 포함된다. 본 발명의 바람직한 실시예에 따르면, 스탬퍼(507)는 에뮬레이터 장치의 초기화와 동시에 동작을 개시하여, 신호 샘플러(502)의 샘플링 기준 클럭이 되는 전술한 클럭 발생 회로(506)에 의해 생성된 클럭 신호(510)에 따라 증가하는 단순한 카운터로 구현될 수 있다.

<41> 다시 도 5를 참조하면, 신호 샘플러(502)에 의해 검출된 핀 신호 값들(509) 및 스탬퍼(507)로부터의 시간 정보(511)는 버퍼(504)에 일시 저장된 후 최종적으로 저장 메모리(505)로 저장된다. 이처럼 저장 메모리(505)에는 샘플링된 핀 신호 뿐

만 아니라 시간 정보인 스탬프 정보도 함께 저장되므로, 후술할 트리거 사건 감지기(503)로부터 출력되는 트리거 발생 시점에서의 스탬프 정보(808) (도 8 참조)를 이용하면 저장 메모리(505)에서 트리거 시점과 관련된 핀 신호의 저장 위치와 시간적 선후 관계를 정확히 식별해 낼 수 있게 된다. 버퍼(504)는 신호 샘플러(502)에 의한 핀 신호의 검출 속도와 저장 메모리(505)의 참조 속도가 다를 경우 이 시차를 보상하기 위한 것으로 바람직하게는 저장 메모리(505)에 저장될 정보의 양을 줄이기 위한 압축 기능도 내장할 수 있다. 저장 메모리(505)는 버퍼(504)로부터 공급되는 신호들을 입력 순서에 따라 차례로 저장하되, 새로이 공급되는 신호들을 저장하기 위해 최초 저장 내용부터 차례로 삭제해 나가는 소위 환형 FIFO(Circular First In First Out) 구조로 구현하는 것이 바람직하다. 또한, 저장 메모리(505)는, 예를 들어 SRAM 등과 같은 고속 메모리로 구현될 수 있으며 확장 가능한 구조로 되는 것이 바람직하다.

<42> 한편, 신호 샘플러(502)로부터 샘플링된 핀 신호들은 트리거 사건 감지기(503)로도 입력된다. 트리거 사건 감지기(503)는 이와 같이 샘플링된 소정 조합의 핀 신호 값들이 사용자에게 의해 미리 설정된 트리거 조건과 일치하는지의 여부를 검사하고, 일치하는 것으로 판단될 경우 그에 따른 인터럽트 요청 신호를 생성하는 역할을 한다. 도 8을 참조하면, 본 발명의 바람직한 실시예에 따른 트리거 사건 감지기(503)의 세부 구성이 예시되어 있다. 도시된 바와 같이, 트리거 사건 감지기(503)는, 특정 신호 형태에 따라 사건 신호(805a, 805b)를 생성하는 복수개의 트리거 유닛 블록(800a, 800b)과, 이들 각 트리거 유닛 블록(800a, 800b)들이 생성하는 복수개의 사건 신호(805a, 805b)들의 선후 관계를 판단하여 그 선후 관계가 미리 지정된 선후 관계와 일치할 경우 최종 트리거 신호(807)를 생성하는 트리거 제어기(806)로 구성된다.

<43> 각 트리거 유닛 블록(800a, 800b)은 샘플링된 다수의 핀 신호들 중 트리거 조건으로 사용되는 핀 신호의 위치를 지정하기 위한 마스크 레지스터(801a), 실제 트리거 조건으로 정의된 각 핀 신호의 패턴 값을 지정하기 위한 패턴 레지스터(802a), 및 이들 각 레지스터의 내용과 샘플링된 입력 데이터(509)를 비트 단위로 비교하는 다수의 논리 게이트(AND 및 XOR 회로)를 포함한다. 마스크 레지스터(801a) 및 패턴 레지스터(802a)에 각각 설정될 트리거 조건 데이터는 후술할 핀 신호 관찰기 제어기(500)의 제어에 의해 사용자가 원하는 값으로 미리 설정된다. 이러한 구성에서는, 샘플링된 각각의 핀 신호 값과 패턴 레지스터(802a)에 설정된 내용이 비트 단위로 서로 비교되는 것으로 되지만, 실제 사건 신호(805a)의 생성시에는 모든 비트위치에서의 이들 핀 신호 값과 패턴 레지스터 내용의 비교 결과가 직접 사용되는 것이 아니라, 마스크 레지스터(801a)의 비트가 1로 설정된 위치에서의 샘플링된 신호 값(509)과 패턴 레지스터(802a)의 해당 비트를 비교한 값만이 유효한 비교 결과 데이터로써 이용되게 된다. 따라서, 이상의 구성은 샘플링된 다수의 핀 신호들중 특정 핀 신호만을 트리거 조건으로 이용하고자 하는 경우에 보다 유용할 수 있다. 그러나, 본 발명에 따른 트리거 사건 감지기(503)는 반드시 전술한 구성에 한정되는 것은 아니고, 마스크 레지스터(801a) 없이 패턴 레지스터(802a)만으로 구성되어, 샘플링된 모든 위치의 핀 신호 값(509)과 패턴 레지스터(802a)의 전체 내용을 비트 단위로 비교하여 그 비교 결과를 사건 신호(805a)의 발생에 사용하도록 구성될 수 있다. 또한, 각각의 트리거 유닛 블록(800a, 800b)은, 도시된 바와 같이 미리 값을 설정(804a)할 수 있는 하향 카운터(803a)를 추가로 포함함으로써, 해당 패턴 일치 사건이 발생하는 횟수를 사용자가 미리 지정하게 하고 전술한 패턴 일치 사건이 이 지정된 횟수 만큼 발생한 경우에 비로서 사건 신호(805a)가 발생하게 할 수도 있다.

<44> 각 트리거 유닛 블록(800a, 800b)으로부터 생성된 사건 신호(805a, 805b)들은 트리거 제어기(806)로 입력되고, 트리거 제어기(806)는 이들 사건 신호(805a, 805b)의 선후 관계가 사용자에게 의해 미리 지정된 선후 관계와 일치하는지의 여부를 조사하여 일치할 경우에만 최종 트리거 신호(807)를 생성한다. 이러한 일련의 사건 신호의 선후 관계의 사전 설정은 트리거 제어기(806)에 상태 머신을 프로그램 함으로써 가능하다.

<45> 한편, 트리거 사건 감지기(503)는 도 5 및 도 8에 도시된 바와 같이, 샘플링된 핀 신호 값(509) 이외에 스탬퍼(507)로부터의 스탬프 정보도 함께 입력받아, 최종 트리거 신호(807)가 발생하는 시점의 해당 스탬프 정보(808)도 같이 출력하도록 함으로써, 프로세싱 엔진(101)측이 해당 트리거 시점을 정확히 식별할 수 있도록 한다. 트리거 사건이 발생한 경우 트리거 제어기(806)에서 생성된 트리거 신호(807)와 트리거 스탬프 정보(808)는 핀 신호 관찰기 제어기(500)로 전달되고, 핀 신호 관찰기 제어기(500)는 통신 채널 제어기(300)를 통해 인터럽트를 프로세싱 엔진(101)으로 전달한다. 프로세싱 엔진이 인터럽트를 받아서 처리하는 과정은 이후에 상세히 설명한다.

<46> 다시 도 5를 참조하면, 핀 신호 관찰기(302) 내의 핀 신호 관찰 제어기(500)는 통신 채널 제어기(300)를 통해 프로세싱 엔진(101)측과 통신하며, 핀 신호 관찰기(302) 내부의 전술한 각각의 기능 블록들을 제어하는 역할을 한다. 즉, 핀 신호 관찰 제어기(500)는 신호 선택기(501)를 제어하여 관찰기(302)로 입력되는 다수의 핀 신호들 중 선택할 신호를 지정하고, 제어신호(512, 513)를 통해 클럭 발생 회로(506)와 스탬퍼(507)를 제어함으로써 신호 샘플러(502)에 의한 핀 신호의 샘플링 동작을 제어하며, 트리거 사건 감지기(503)를 제어하여 트리거 조건을 설정하고, 저장 메모리(505)를 제어하여 저장된 핀 신호 정보를 통신 채널 제어기(300)를 통해 프로세싱 엔진(101)측에 전달하는

역할을 한다.

<47> 이하에서는, 지금까지 상술한 본 발명에 따른 에뮬레이터 장치의 구성과 도 9 내지 11에 도시된 플로우챗트를 이용하여, 타겟 인터페이스 엔진(102)의 핀 신호 관찰기(302)측에서 발생한 하드웨어 트리거 사건과 프로세싱 엔진(101)의 소프트웨어 변수 관찰기(401)측에서 발생한 소프트웨어 트리거 사건의 발생에 응답하여 일어나는 일련의 처리 과정에 대해 설명하기로 한다.

<48> 먼저, 도9는 핀 신호 관찰기(302)의 트리거 사건 감지기(503)에서 하드웨어 트리거 사건 발생에 응답하여 핀 신호 관찰기(302)측에서 일어나는 처리 과정을 도시한 플로우챗트이다. 본 처리 과정은 우선 에뮬레이션 시스템이 초기화되는 단계(901)로부터 시작된다. 에뮬레이션 시스템이 초기화되면 단계(902)에서 핀 신호 관찰기(302) 내부의 시스템퍼(507)가 프로세싱 엔진(101) 내의 타이머(204)와 동기되어 동작을 개시한다. 이어, 단계(903)에서 핀 신호의 샘플링을 개시하라는 상태가 될 때까지 기다린다. 단계(904)에서 핀 신호 생성기(301)와 타겟 시스템(120) 사이의 입출력 핀 신호(304) 및 핀 신호 생성기 내부의 신호(303)들 중 일부가 신호 선택기(501)에 의해 선택되고, 이들은 신호 샘플러(502)에 의해 클럭 발생 회로(506)에서 제공되는 클럭 단위로 샘플링되어 버퍼(504)에 저장된 후 다시 저장 메모리(505)에 순차적으로 저장되는 과정이 반복되어 나간다. 샘플링된 핀 신호들은 단계(905)에서 트리거 사건 감지기(503)에 의해 검사되어 미리 사용자에게 의해 트리거 사건으로 설정된 특정 조건과 일치하는지의 여부를 결정한다. 단계(905)에서 트리거 사건이 발생하지 않은 것으로 판단된 경우에는 단계(904)로 되돌아가 핀 신호의 샘플링 및 저장 동작을 계속하여 수행해 나가고, 반면 트리거 사건 감지기(503)에서 트리거 사건이 발생한 것으로 판단된 경우에는 단계(906)로 진행하여

핀 신호의 샘플링 및 저장 동작을 중지하고, 계속하여 단계(907)에서 프로세싱 엔진(101)측으로 인터럽트 요청 신호를 발생시킨다. 한편, 프로세싱 엔진(101)측에서는 이 인터럽트 요청 신호를 수신하게 되면 소프트웨어 변수값 및 핀 신호 값에 대한 분석을 행하고, 분석 종료시 핀 신호 관찰기(302)를 재기동시키는 신호를 핀 신호 관찰기(302)측으로 전송한다 (이에 대해서는, 이하 도 10을 참조하여 상술하기로 한다). 단계(908)에서는, 핀 신호 관찰기(302)가 이와 같은 재기동 신호를 프로세싱 엔진(101)측으로부터 수신하였는지를 판단하여, 재기동 신호 수신시, 단계(903)으로 되돌아가 핀 신호의 샘플링 동작을 재개한다.

<49> 도 10의 플로우챠트를 참조하여, 도 9에서 설명된 핀 신호 관찰기 측에서 감지된 하드웨어 트리거 사건에 따라 발생한 인터럽트 요청 신호에 의해 프로세싱 엔진(101) 측에서 일어나는 처리 과정에 대해 설명하기로 한다. 우선, 단계(1000)에서 에뮬레이션 시스템이 초기화되어 동작을 개시하면, 프로세싱 엔진(101)의 프로세서(201)는 메모리(202) 내에 저장된 타겟 칩 알고리즘(400)을 실행해 나간다. 단계(1001)에서는, 알고리즘(400)의 실행 도중에 핀 신호 관찰기(302)로부터 인터럽트 요청 신호가 발생했는지의 여부를 점검한다. 만일 인터럽트 요청 신호가 수신된 경우에는, 단계(1002)에서 핀 신호 관찰기(302)측의 저장 메모리(505)에 저장된 핀 신호의 내용을 메모리(202)에 할당된 핀 신호값 저장 버퍼(405)로 복사한다. 계속하여, 단계(1003)에서, 프로세서(201)는 메모리(202)내의 변수 및 신호 분석기(406)를 실행하여 소프트웨어 변수값 저장 버퍼(404)와 핀 신호값 저장 버퍼(405)의 내용을 이용하여 필요한 디버깅을 수행한다. 마지막으로, 단계(1004)에서는, 핀 신호 관찰기(302)측으로 재기동 신호를 전송하여 핀 신호 관찰기(302)를 다시 기동시킴으로써, 다음으로 발생할 트리거 사건에 대비한다.

<50> 다음으로, 도 11을 참조하여, 타겟 칩 알고리즘 내의 소프트웨어 트리거 사건 발생에 응답하여 프로세싱 엔진(101)측에서 일어나는 처리 과정에 대해 설명하기로 한다.

우선, 단계(1101)에서 에뮬레이션 장치가 초기화되어 동작을 개시하면, 단계(1102)에서 타이머(204)를 초기화한다. 이 때, 도9에서 설명된 바와 같이, 핀 신호 관찰기(302) 내의 8스탬퍼(507) 역시 동기되어 초기화된다. 이어서, 단계(1103)에서는 타겟 칩 알고리즘(400)의 실행을 진행해 나가고, 단계(1104)에서 알고리즘의 실행이 종료된 것으로 판단되면, 단계(1105)로 진행하여 처리를 종료한다. 한편, 단계(1104)에서 알고리즘(400)이 아직 완료되지 않은 것으로 판단된 경우에는, 단계(1106)에서 프로세서(201)가 타겟 칩 알고리즘(400) 내에 삽입된 소프트웨어 변수 관찰기 함수 루틴을 만나게 되었는지의 여부를 판단하고, 만일 소프트웨어 변수 관찰기 루틴이 아직 나타나지 않은 것으로 판단된 경우에는 단계(1103)로 되돌아가 타겟 칩 알고리즘(400)을 계속 실행해 나가고 이 과정은 소프트웨어 변수 관찰기 루틴이 나타날 때까지 계속 반복된다. 한편, 단계(1106)에서 프로세서(201)가 소프트웨어 변수 관찰기 루틴을 만난 경우에는, 단계(1107)에서 해당 소프트웨어 변수 관찰기 루틴을 실행함으로써 이에 의해 관찰 대상이 되고 있는 소프트웨어 변수의 값을 메모리(202) 내의 소프트웨어 변수값 저장 버퍼(404)에 저장한다. 계속하여, 단계(1108)에서는, 소프트웨어 관찰기 루틴에 의해 관찰된 소프트웨어 변수값을 기초로 소프트웨어 트리거 사건이 발생했는지의 여부를 판단한다. 트리거가 발생하지 않은 것으로 판단된 경우에는 단계(1103)로 되돌아가 남은 부분의 타겟 칩 알고리즘을 계속 실행하고, 반면 관찰된 소프트웨어 변수 값이 트리거 조건과 일치하는 경우에는 단계(1109)에서 핀 신호 관찰기(302)에 의한 핀 신호의 샘플링 동작을 정지시킨 다음, 계속하여, 단계(1110)에서 그 순간 핀 신호 관찰기(302)측의 저장 메모리(505)에

저장된 핀 신호의 내용을 불러와 이를 메모리(202) 내의 핀 신호 값 저장 버퍼(405)에 복사한다. 최종적으로 단계(1111)에서는, 메모리(202) 내의 변수 및 신호 분석기(406)를 실행하여 소프트웨어 변수값 저장 버퍼(404)와 핀 신호값 저장 버퍼(405)에 각각 저장된 내용을 이용하여 필요한 디버깅을 수행한 다음, 단계(1103)로 되돌아가 타겟 칩 알고리즘(400)의 남은 부분을 계속 실행해 간다.

<51> 이하에서는, 도 12 및 13을 참조하여, 이상 설명한 본 발명에 따른 디버깅 방법을 통해 프로세싱 엔진측과 타겟 인터페이스 엔진측에서 각각 관찰되는 하드웨어 핀 신호 값 및 소프트웨어 변수 값이 손실 없이 저장되는 원리에 대해 설명하기로 한다.

<52> 먼저, 도 12의 타이밍도를 참조하여, 프로세싱 엔진과 타겟 인터페이스 엔진으로 구성된 종래의 에뮬레이터 장치에 있어서, 프로세싱 엔진측에서 측정되는 소프트웨어 변수 값과 타겟 인터페이스 엔진측의 핀 신호 관찰기에서 관찰되는 하드웨어 핀 신호 값 사이에 측정 지연이 발생하게 되는 이유를 설명한다. 에뮬레이션 수행 도중에 사용자에 의해 설정된 핀 신호의 조합으로 정의되는 특정 트리거 사건(1201)이 발생하게 되면, 핀 신호 관찰기는 내부의 트리거 사건 감지기를 통해 이를 감지하고, 프로세싱 엔진내의 프로세서로 트리거 인터럽트 요청 신호(1202)를 전달한다. 동시에, 핀 신호 관찰기는 트리거 사건 발생 이후 관찰되는 모든 하드웨어 핀 신호의 변화(1203)를 핀 신호 저장 메모리에 저장해 나간다. 한편, 프로세싱 엔진측에서는, 트리거 요청 신호가 프로세서로 전달되어 이에 대한 응답 신호(1204)의 송신이 있을 후, 소프트웨어 변수 값과 하드웨어 핀 신호 값의 동기를 맞추기 위한 타이머가 동작한 이후(1205)에야 비로서 하드웨어 트리거 사건 발생과 연관된 소프트웨어 변수의 변화(1206)를 소프트웨어 변수값 저장 버퍼에 저장하기 시작한다. 그러나, 전술한 트리거 인터럽트 요청 신호(1201)의 전송에는

하드웨어적인 시간 지연이 필연적으로 수반될 뿐만 아니라, 실제로 인터럽트 요청 신호(1201)가 프로세서에 도달한 경우라 하더라도, 그 순간 프로세서가 관찰 대상 소프트웨어 에뮬레이션 알고리즘 내에 삽입된 소프트웨어 변수 관찰기 루틴을 수행하고 있는 중이라면, 프로세서는 수신한 트리거 인터럽트 요청 신호에 즉시 응답할 수 없고 현재 수행 중인 소프트웨어 변수 관찰기 루틴을 모두 수행한 뒤에야 비로서 요청 신호(1201)에 대한 트리거 인터럽트 응답(1204)을 할 수 있게 된다. 따라서, 이와 같은 하드웨어 핀 신호와 소프트웨어 변수 사이에 존재하는 하드웨어적인 시간 지연 및 소프트웨어적인 시간 지연의 인터럽트 응답 대기 시간으로 인해, 만일 인터럽트 응답 대기 시간 내에서 소프트웨어 변수 값이 변화하는 경우에는 그 변화 데이터를 저장할 수 없게 되어 버린다.

핀 신호 관찰기측과 프로세싱 엔진측이 통신 채널로서, 예를 들어 PCI 버스에 의해 상호 연결된 경우에는, 실제로 이러한 인터럽트 응답 대기 시간은 수 백 내지는 수 천 PCI 싸이클 정도 소요되어 데이터의 손실이 심각한 정도에 이른다.

<53> 도 13은 핀 신호 관찰기에 의한 하드웨어 트리거 사건의 감지에 응답하여, 본 발명에 따라 관찰 대상 소프트웨어 변수 값이 저장되는 과정을 설명한 도면이다. 도 13의 횡축은 시간 축을, 도형 (마름모, 원형)은 소프트웨어 변수의 변화를 각각 나타내고, 도면 하단의 변수 기록 창은 각 시간에서의 소프트웨어 변수값 저장 버퍼에 저장된 변수 변화의 내용을 나타낸다. 즉, 변수 기록 창(A)은 시간 A에서의 소프트웨어 변수의 변화 데이터가 저장된 저장 버퍼의 저장 내용을, 변수 기록 창(B)은 시간 B에서의 소프트웨어 변수의 변화 데이터가 저장된 저장 버퍼의 저장 내용을, 변수 기록 창(C)은 시간 C에서의 소프트웨어 변수의 변화 데이터가 저

장된 저장 버퍼의 저장 내용을 각각 나타낸다. 이 때, 소프트웨어 변수값 저장 버퍼의 용량, 즉 변수 기록 창의 크기는 전술한 인터럽트 응답 대기 시간 동안의 변수 변화 데이터를 기록할 수 있을 정도의 크기로 설정되는 것이 바람직하다. 이와 같이, 본 발명에 따르면, 소정 시간 단위마다 (보다 구체적으로는, 프로세서가 관찰 대상 소프트웨어 애플리케이션 알고리즘내에 삽입된 소프트웨어 변수 관찰기 루틴을 실행할 때마다) 발생하는 소프트웨어 변수의 변화 데이터를 적어도 인터럽트 응답 대기 시간 동안 소프트웨어 변수값 저장 버퍼가 항상 갱신 저장하고 있기 때문에, 핀 신호 관찰기측에서의 트리거 사건 발생 감지(1301)로부터 이에 대한 인터럽트 응답(1302)이 일어나기까지 인터럽트 응답 대기 시간이 존재하더라도 이 기간 중에 변화하는 소프트웨어 변수 값을 손실 없이 저장할 수 있게 된다.

<54> 비록 이상의 도 12 및 13를 참조한 설명에서는 핀 신호 관찰기측에 의해 하드웨어 트리거 사건이 감지될 경우의 프로세싱 엔진측에서 발생하는 소프트웨어 변수 값 측정 지연과 관련된 본 발명에 따른 동작 원리를 설명하였으나, 도 12 및 13의 설명은 전술한 도 9, 10, 11과 관련한 설명 부분에서와 같이, 프로세싱 엔진측에 의한 소프트웨어 트리거 사건의 감지시에 핀 신호 관찰기측에서 발생하는 하드웨어 핀 신호 값의 측정 지연에 대해서도 유사하게 적용될 수 있으며, 이에 따른 하드웨어 핀 신호 값의 데이터 손실 역시도 방지할 수 있게 된다.

【발명의 효과】

<55> 이상 설명한 바와 같이, 본 발명에 따르면, 프로세싱 엔진과 타겟 인터페이스 엔진이 상호 작용하는 에뮬레이션 환경에서, 타겟 인터페이스 엔진측에서 관찰

되는 하드웨어 핀 신호 값과 프로세싱 엔진측에서 실행 중인 소프트웨어 변수의 값의 관찰 타이밍을 동기시키는 것이 가능하게 되고, 이에 따라 타겟 시스템을 포함한 전체 시스템의 검증 시간을 줄일 수 있게 된다.

<56> 이상 설명한 본 발명의 기술 사상과 범주를 벗어나지 않는 범위에서, 본 기술 분야의 당업자라면, 상술한 실시예 이외에도 다양한 변경 및 수정 실시가 가능함을 알 수 있을 것이다. 따라서, 본 발명의 범위는 상술한 상세한 설명의 기재에 의해 제한적으로 해석되어서는 안되며, 오로지 이하의 특허청구범위에 의해서만 그 범위가 정해져야 한다.

【특허청구범위】**【청구항 1】**

타겟 시스템에 장착될 타겟 칩의 논리 설계를 검증하기 위한 에뮬레이터 장치에 있어서,

상기 타겟 칩의 논리 설계에 대응하는 적어도 하나의 소프트웨어 변수를 포함하는 소프트웨어 알고리즘을 실행하기 위한 프로세싱 엔진, 및

상기 타겟 시스템과 인터페이스하고 있으며, 상기 프로세싱 엔진의 알고리즘 처리에 응답하여 상기 타겟 시스템으로 또는 상기 타겟 시스템으로부터 핀 신호를 송수신하기 위한 타겟 인터페이스 엔진

을 포함하고,

상기 적어도 하나의 소프트웨어 변수와 상기 핀 신호는 상기 알고리즘의 실행에 따라 시간적으로 변화하며, 상기 프로세싱 엔진은 선정된 시간에서의 상기 적어도 하나의 소프트웨어 변수 값과 상기 핀 신호 값을 대응시키기 위한 수단을 포함하는 에뮬레이터 장치.

【청구항 2】

제1항에 있어서, 상기 타겟 인터페이스 엔진은 i)상기 타겟 시스템으로 송신될 핀 신호를 생성하며 상기 타겟 시스템으로부터 핀 신호를 수신하기 위한 핀 신호 생성기와, ii) 상기 핀 신호 생성기와 상기 타겟 시스템 사이에서 송수신되는 상기 핀 신호를 입력받아 저장하기 위한 핀 신호 관찰기를 포함하는 에뮬레이터 장치.

【청구항 3】

제2항에 있어서, 상기 편 신호 관찰기에 저장되는 편 신호는 상기 편 신호와 연관된 시간 정보와 함께 저장되는 에뮬레이터 장치.

【청구항 4】

제3항에 있어서, 상기 편 신호 관찰기는 상기 편 신호 값이 특정 조건을 만족시키는 경우를 정의하는 제1 트리거 사건의 발생을 감지하기 위한 트리거 사건 감지기를 포함하는 에뮬레이터 장치.

【청구항 5】

제4항에 있어서, 상기 트리거 사건 감지기는 상기 제1 트리거 사건의 발생에 응답하여 상기 프로세싱 엔진으로 제1 인터럽트 요청 신호를 전송하기 위한 수단을 더 포함하는 에뮬레이터 장치.

【청구항 6】

제5항에 있어서, 상기 선정된 시간은 상기 제1 트리거 사건의 발생 시점과 시간적으로 관련되어 있는 에뮬레이터 장치.

【청구항 7】

제1항 또는 제6항에 있어서, 상기 프로세싱 엔진은 상기 시간에 따라 변화하는 적어도 하나의 소프트웨어 변수를 관찰하고, 관찰되는 소프트웨어 변수 값이 저장될 수 있도록 제어하기 위한 소프트웨어 변수 관찰기를 포함하는 에뮬레이터 장치.

【청구항 8】

제7항에 있어서, 상기 소프트웨어 변수 관찰기는 상기 소프트웨어 알고리즘내에 삽입되는 소프트웨어 루틴인 에뮬레이터 장치.

【청구항 9】

제7항에 있어서, 상기 프로세싱 엔진은 상기 관찰되는 소프트웨어 변수 값을 저장하기 위한 소프트웨어 변수값 저장 버퍼를 더 포함하는 에뮬레이터 장치.

【청구항 10】

제9항에 있어서, 상기 소프트웨어 변수값 저장 버퍼에 저장되는 소프트웨어 변수값은 그와 연관된 시간 정보와 함께 저장되는 에뮬레이터 장치.

【청구항 11】

제10항에 있어서, 상기 프로세싱 엔진은 상기 인터럽트 요청 전송 수단으로부터의 제1 인터럽트 요청 신호의 전송 시점으로부터 소정의 응답 대기 시간 경과 후 응답하며, 상기 소프트웨어 변수값 저장 버퍼는 최소한 상기 소정의 응답 대기 시간 동안 상기 관찰되는 소프트웨어 변수의 시간적 변화값을 저장하기에 충분한 저장 용량을 가지는 에뮬레이터 장치.

【청구항 12】

제10항에 있어서, 상기 프로세싱 엔진은 핀 신호값 저장 버퍼 및, 상기 제1 인터럽트 요청 신호에 응답하여 상기 핀 신호 관찰기에 저장된 핀 신호 값을 상기 핀 신호값 저장 버퍼에 저장하는 것을 제어하기 위한 수단을 더 포함하는 에뮬레이터 장치.

【청구항 13】

제9항에 있어서, 상기 소프트웨어 변수 관찰기는 상기 시간에 따라 변화하는 소프트웨어 변수의 값이 특정 조건을 만족시키는 경우를 정의하는 제2 트리거 사건의 발생을 감지하기 위한 수단을 더 포함하는 에뮬레이터 장치.

【청구항 14】

제13항에 있어서, 상기 소프트웨어 변수 관찰기는 상기 제2 트리거 사건의 발생에 응답하여 제2 인터럽트 요청 신호를 출력하기 위한 수단을 포함하는 에뮬레이터 장치.

【청구항 15】

제14항에 있어서, 상기 프로세싱 엔진은 상기 제2 인터럽트 요청 신호에 응답하여 상기 핀 신호 관찰기에 저장된 핀 신호 값을 상기 핀 신호값 저장 버퍼에 저장하는 것을 제어하기 위한 수단을 포함하는 에뮬레이터 장치.

【청구항 16】

제12항 또는 제15항에 있어서, 상기 프로세싱 엔진은 상기 소프트웨어 변수값 저장 버퍼 및 상기 핀 신호값 저장 버퍼에 저장된 정보에 근거하여 상기 에뮬레이터 장치의 에뮬레이션 상태 또는 오류를 분석하기 위한 변수 및 신호 분석기를 더 포함하는 에뮬레이터 장치.

【청구항 17】

타겟 시스템에 장착될 타겟 칩의 논리 설계를 검증하기 위한 에뮬레이터 장치에 있어서,

상기 타겟 칩의 논리 설계에 대응하는 적어도 하나의 소프트웨어 변수를 포함하는 소프트웨어 알고리즘을 실행하기 위한 프로세싱 엔진, 및

상기 타겟 시스템과 인터페이스하고 있으며, 상기 프로세싱 엔진의 알고리즘 처리에 응답하여 상기 타겟 시스템으로 또는 상기 타겟 시스템으로부터 핀 신호를 송수신하기 위한 타겟 인터페이스 엔진을 포함하고,

상기 적어도 하나의 소프트웨어 변수와 상기 핀 신호는 상기 알고리즘의 실행에 따라 시간적으로 변화하며, 상기 프로세싱 엔진은 상기 적어도 하나의 소프트웨어 변수의 시간에 따른 변화를 추적하기 위한 제1 수단을 포함하며, 상기 타겟 인터페이스 엔진은 상기 핀 신호의 시간에 따른 변화를 추적하기 위한 제2 수단을 포함하며, 상기 제1 수단과 상기 제2 수단은 동작적으로 결합되어 있는 에뮬레이터 장치.

【청구항 18】

제17항에 있어서, 상기 타겟 인터페이스 엔진은 상기 타겟 시스템으로 송신될 핀 신호를 생성하며 상기 타겟 시스템으로부터 핀 신호를 수신하기 위한 핀 신호 생성기를 더 포함하며, 상기 제2 수단은 상기 핀 신호 생성기와 상기 타겟 시스템 사이에서 송수신되는 상기 핀 신호를 입력받아 저장하기 위한 핀 신호 관찰기를 포함하는 에뮬레이터 장치.

【청구항 19】

제18항에 있어서, 상기 핀 신호 관찰기에 저장되는 핀 신호는 상기 핀 신호와 연관된 시간 정보와 함께 저장되는 에뮬레이터 장치.

【청구항 20】

제19항에 있어서, 상기 핀 신호 관찰기는 상기 핀 신호 값이 특정 조건을 만족시키는 경우를 정의하는 제1 트리거 사건의 발생을 감지하기 위한 트리거 사건 감지기를 포함하는 에뮬레이터 장치.

【청구항 21】

제20항에 있어서, 상기 트리거 사건 감지기는 상기 제1 트리거 사건의 발생에 응답하여 상기 프로세싱 엔진으로 제1 인터럽트 요청 신호를 전송하기 위한 수단을 더 포함하는 에뮬레이터 장치.

【청구항 22】

제17항 또는 제21항에 있어서, 상기 제1 수단은 상기 시간에 따라 변화하는 적어도 하나의 소프트웨어 변수를 관찰하고, 관찰되는 소프트웨어 변수 값이 저장될 수 있도록 제어하기 위한 소프트웨어 변수 관찰기를 포함하는 에뮬레이터 장치.

【청구항 23】

제22항에 있어서, 상기 소프트웨어 변수 관찰기는 상기 소프트웨어 알고리즘내에 삽입되는 소프트웨어 루틴인 에뮬레이터 장치.

【청구항 24】

제22항에 있어서, 상기 제1 수단은 상기 관찰되는 소프트웨어 변수 값을 저장하기 위한 소프트웨어 변수값 저장 버퍼를 더 포함하는 에뮬레이터 장치.

【청구항 25】

제24항에 있어서, 상기 소프트웨어 변수값 저장 버퍼에 저장되는 소프트웨어 변수값은 그와 연관된 시간 정보와 함께 저장되는 에뮬레이터 장치.

【청구항 26】

제25항에 있어서, 상기 제1 수단은 상기 인터럽트 요청 전송 수단으로부터의 제1 인터럽트 요청 신호의 전송 시점으로부터 소정의 응답 대기 시간 경과 후 응답하며, 상기 제1 수단은 상기 소프트웨어 변수값 저장 버퍼는 최소한 상기 소정의 응답 대기 시간 동안 상기 관찰되는 소프트웨어 변수의 시간적 변화값을 저장하기에 충분한 저장 용량을 가지는 에뮬레이터 장치.

【청구항 27】

제25항에 있어서, 상기 제1 수단은 핀 신호값 저장 버퍼, 및 상기 제1 인터럽트 요청 신호에 응답하여 상기 핀 신호 관찰기에 저장된 핀 신호 값을 상기 핀 신호값 저장 버퍼에 저장하는 것을 제어하기 위한 수단을 더 포함하는 에뮬레이터 장치.

【청구항 28】

제24항에 있어서, 상기 소프트웨어 변수 관찰기는 상기 시간에 따라 변화하는 소프트웨어 변수의 값이 특정 조건을 만족시키는 경우를 정의하는 제2 트리거 사건의 발생을 감지하기 위한 수단을 더 포함하는 에뮬레이터 장치.

【청구항 29】

제28항에 있어서, 상기 소프트웨어 변수 관찰기는 상기 제2 트리거 사건의 발생에 응답하여 제2 인터럽트 요청 신호를 출력하기 위한 수단을 포함하는 에뮬레이터 장치.

【청구항 30】

제29항에 있어서, 상기 제1 수단은 상기 제2 인터럽트 요청 신호에 응답하여 상기 핀 신호 관찰기에 저장된 핀 신호 값을 상기 핀 신호값 저장 버퍼에 저장하는 것을 제어하기 위한 수단을 포함하는 에뮬레이터 장치.

【청구항 31】

제27항 또는 제30항에 있어서, 상기 제1 수단은 상기 소프트웨어 변수값 저장 버퍼 및 핀 신호값 저장 버퍼에 저장된 정보에 근거하여 상기 에뮬레이터 장치의 에뮬레이션 상태 또는 오류를 분석하기 위한 변수 및 신호 분석기를 더 포함하는 에뮬레이터 장치.

【청구항 32】

타겟 시스템에 장착될 타겟 칩의 논리 설계를 검증하기 위한 에뮬레이션 방법에 있어서,

상기 타겟 칩의 논리 설계에 대응하는 적어도 하나의 소프트웨어 변수를 포함하는 소프트웨어 알고리즘을 실행하는 단계 - 상기 적어도 하나의 소프트웨어 변수는 상기 소프트웨어 알고리즘의 실행에 따라 시간적으로 변화함 -,

상기 소프트웨어 알고리즘의 실행에 따라 상기 타겟 시스템으로 전송될 핀 신호를 생성하는 단계 - 상기 핀 신호는 상기 소프트웨어 알고리즘의 실행에 따라 시간적으로 변화함 -, 및

상기 적어도 하나의 소프트웨어 변수와 상기 핀 신호의 변화 값에 대한 시간적 이력을 제공하는 단계

를 포함하는 에뮬레이션 방법.

【청구항 33】

제32항에 있어서, 상기 시간적 이력을 제공하는 단계는 상기 적어도 하나의 소프트웨어 변수의 시간에 따른 변화를 기록하는 단계를 포함하는 에뮬레이션 방법.

【청구항 34】

제32항에 있어서, 상기 시간적 이력을 제공하는 단계는 상기 핀 신호의 시간에 따른 변화를 기록하는 단계를 포함하는 에뮬레이션 방법.

【청구항 35】

제34항에 있어서, 상기 핀 신호 값이 특정 조건을 만족시키는 경우를 정의하는 제1 트리거 사건의 발생을 감지하는 단계를 더 포함하는 에뮬레이션 방법.

【청구항 36】

제35항에 있어서, 상기 제1 트리거 사건의 발생에 응답하여 제1 인터럽트 요청 신호를 발생시키는 단계를 더 포함하는 에뮬레이션 방법.

【청구항 37】

제33항에 있어서, 상기 시간에 따라 변화하는 소프트웨어 변수의 값이 특정 조건을 만족시키는 경우를 정의하는 제2 트리거 사건의 발생을 감지하는 단계를 더 포함하는 에뮬레이션 방법.

【청구항 38】

제37항에 있어서, 상기 제2 트리거 사건의 발생에 응답하여 제2 인터럽트 요청 신호를 발생시키는 단계를 더 포함하는 에뮬레이션 방법.

【청구항 39】

제36항 또는 제38항에 있어서, 상기 제1 인터럽트 요청 신호 및 상기 제2 인터럽트 요청 신호 중의 어느 하나의 발생에 응답하여 상기 제공된 시간적 이력을 참조하는 단계를 더 포함하는 에뮬레이션 방법.

【청구항 40】

타겟 시스템에 장착될 타겟 칩의 논리 설계를 검증하기 위한 에뮬레이터 장치에 있어서,

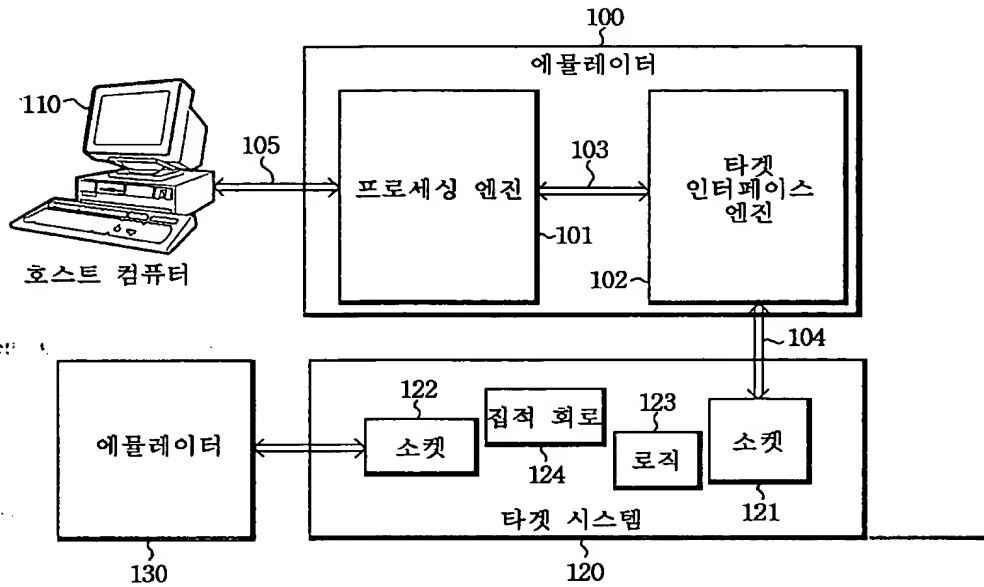
상기 타겟 칩의 논리 설계에 대응하는 적어도 하나의 소프트웨어 변수를 포함하는 소프트웨어 알고리즘을 실행하여 상기 타겟 시스템으로 또는 상기 타겟 시스템으로부터 편 신호를 송수신하기 위한 수단 -상기 적어도 하나의 소프트웨어 변수와 상기 편 신호는 상기 알고리즘의 실행에 따라 시간적으로 변화함-, 및

상기 소프트웨어 변수의 시간에 따른 변화와 상기 편 신호의 시간에 따른 변화를 적어도 소정의 시간 동안 기록하기 위한 수단

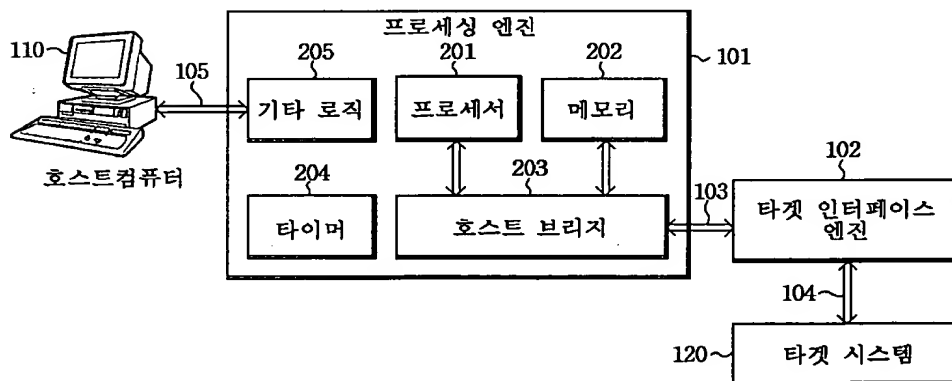
을 포함하는 에뮬레이터 장치.

【도면】

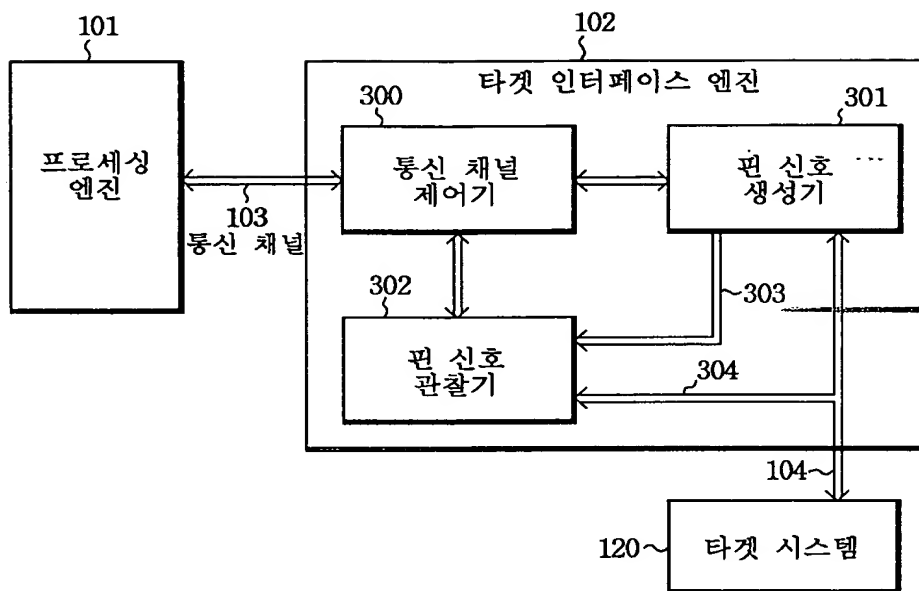
【도 1】



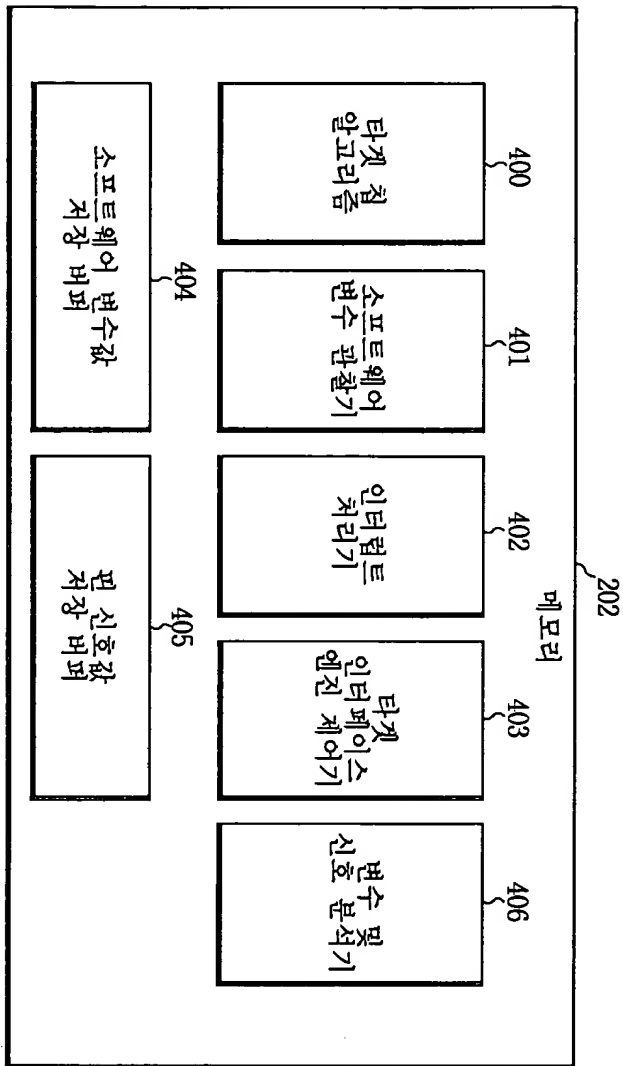
【도 2】



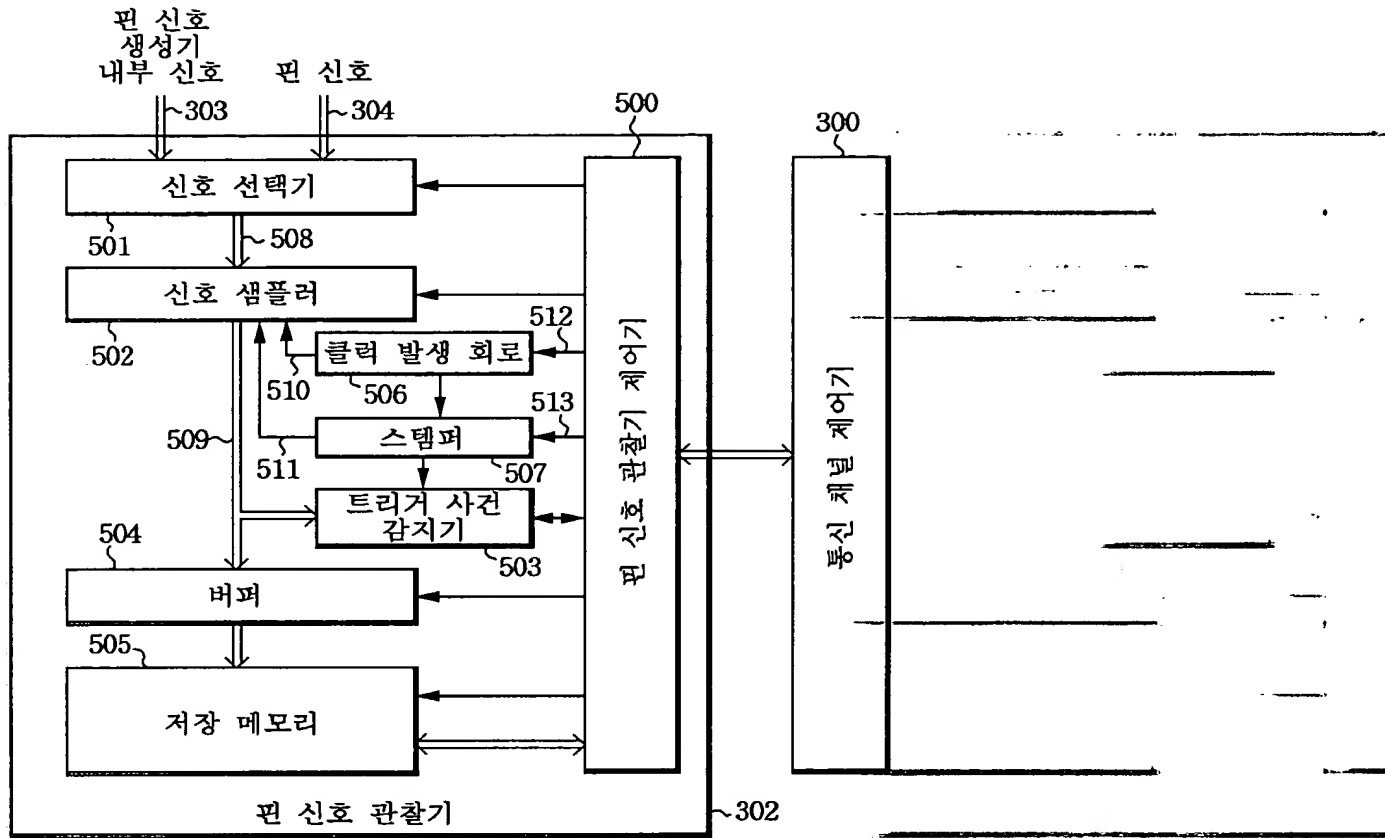
【도 3】



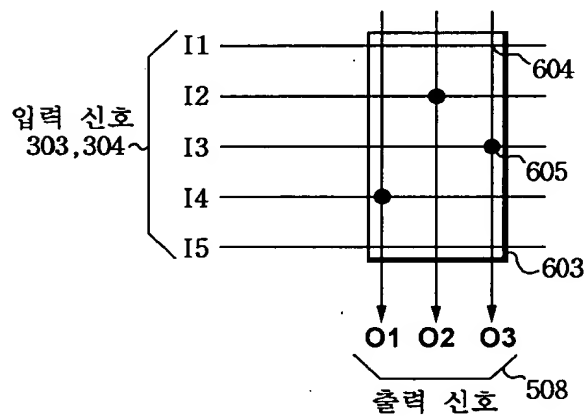
【도 4】



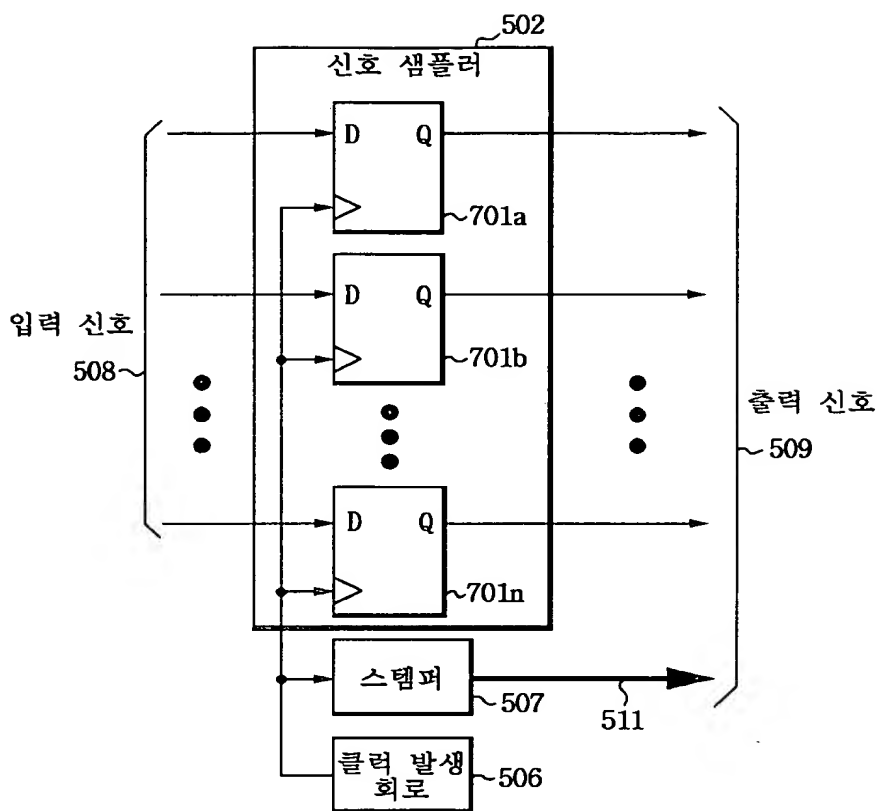
【도 5】



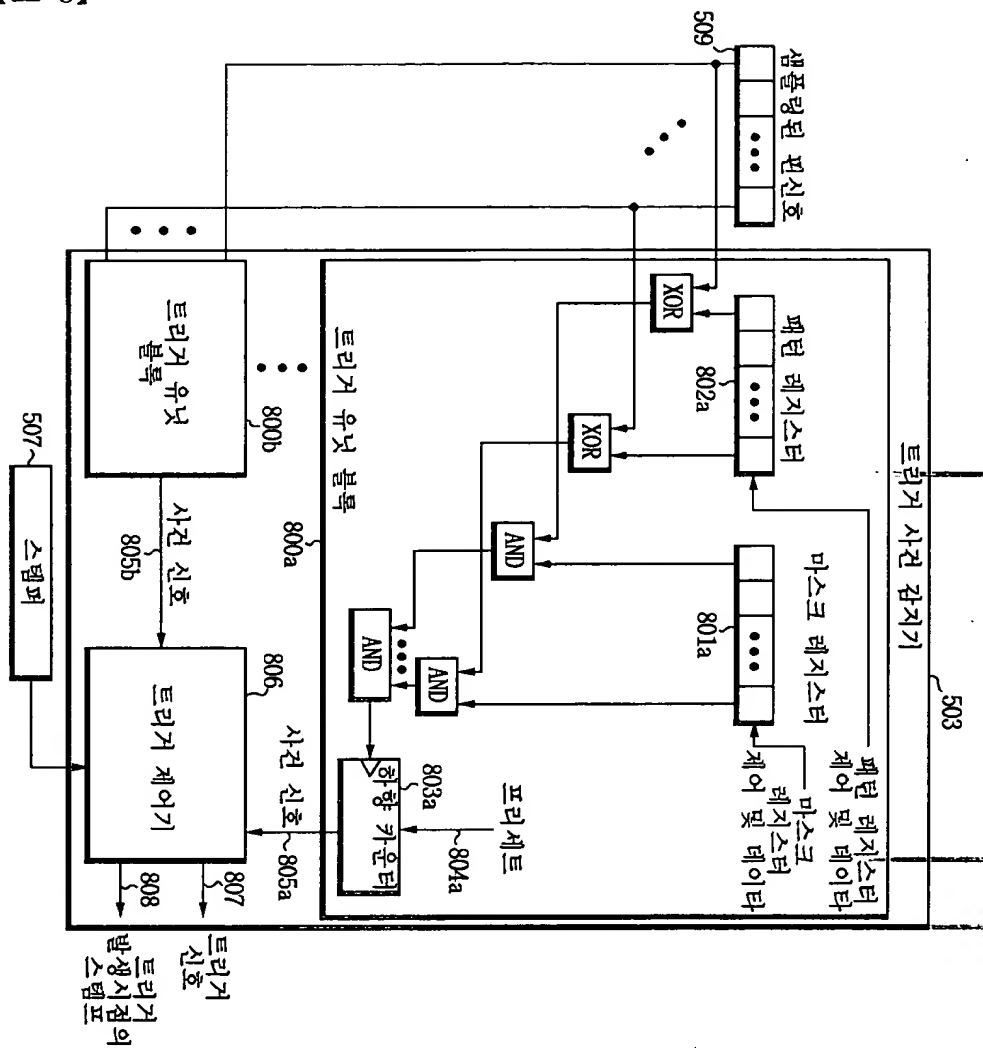
【도 6】



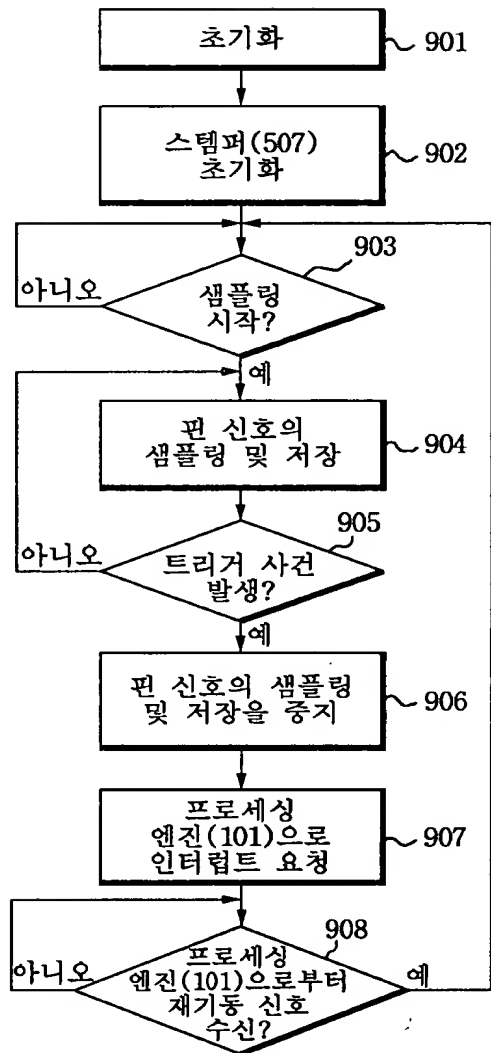
【도 7】



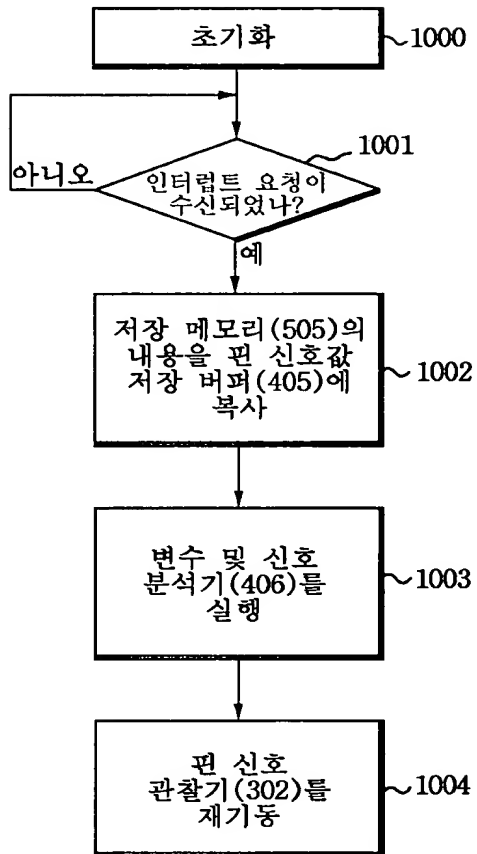
【도 8】



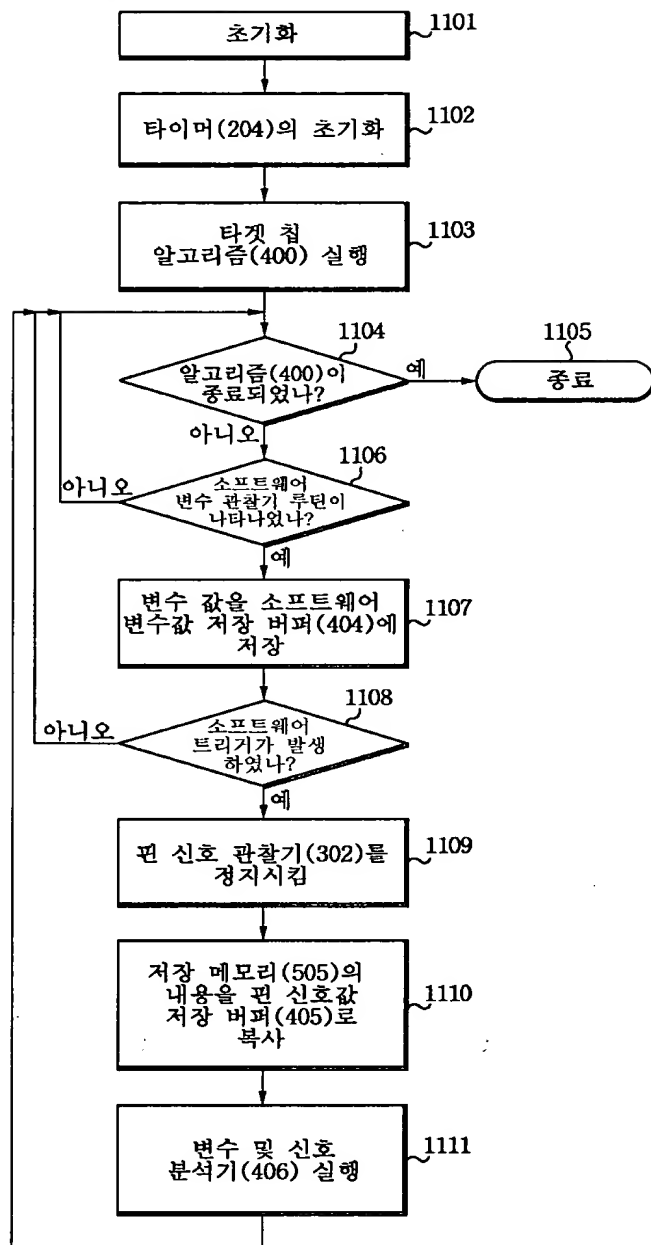
【도 9】



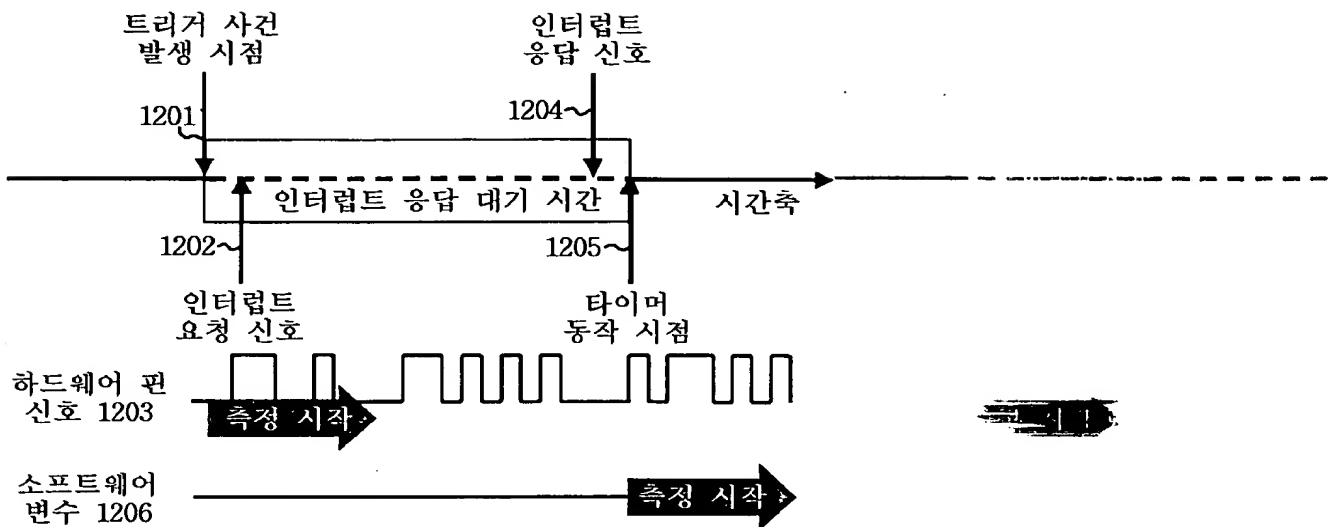
【도 10】



【도 11】



【도 12】



【도 13】

